```
local _ = require"lib"
local any,big,cat,chat,cli,coerce    = _.any,_.big,_.cat,_.chat,_.cli,_.coerce
local csv,fmt,get,gt                 = _.csv,_.fmt,_.get,_.gt
local klass,lines,lt,main,many,map   = _.klass,_.lines,_.lt,_.main,_.many,_.map
local obj,per,push,rand,rev,rnd      = _.obj,_.per,_.push,_.rand,_.rev,_.rnd
local rogues,same,shuffle,slice,sort = _.rogues,_.same,_.shuffle,_.slice,_.sort
local values,words                   = _.values,_.words

local _ = require"xai"
local the, help,csv2data =  _.the, _.help, _.csv2data
local ABOUT,DATA,NOM     = _.ABOUT, _.DATA,_.NOM
local RATIO,ROW,XY       = _.RATIO,_.ROW,_.XY
local bins,half,how      = _.bins, _.half, _.how

---- ---- ---- ---- Tests
-- Tests fail if they do not return `true`.
local go={}

function go.pass() return true end

function go.the() chat(the); return true end

function go.nom(   nom)
  nom=NOM()
  for i=1,1 do
    for _,x in pairs{"a","a","a","a","b","b","c"} do
      nom:add(x) end end
  return "a"==nom:mid() and 1.38==rnd(nom:div(),2) end

function go.ratio(    r)
  r=RATIO()
  the.ratios = 64
  for i=1,100 do r:add(i) end
  return 52==r:mid() and 32.56==rnd(r:div(),2)  end

function go.about()
  map(  ABOUT{"Clndrs","Volume","Hp:","Lbs-",
          "Acc+","Model","origin","Mpg+"}.y , chat)
  return true end

function go.one(     data1,data2)
  data1=csv2data("../data/auto93.csv")
  print("mid1", cat(data1:mid(2)))
  print("div1", cat(data1:div(2)))
  data2=            data1:clone(data1.rows)
  print("mid2", cat(data2:mid(2)))
  print("div2", cat(data2:div(2)))
  return true
  end

function go.dist(    data,row1,row2)
  data= csv2data("../data/auto93.csv")
  print(#data.rows)
  for i = 1,20 do
    row1=any(data.rows)
    row2=any(data.rows)
    print(row1-row2) end
  return true end

function go.betters(   data,data1,data2)
  data= csv2data("../data/auto93.csv")
  data.rows = sort(data.rows)
  data1=data:clone(slice(data.rows,1,50))
  data2=data:clone(slice(data.rows,(#data.rows)-50))
  map({data1:mid(), data2:mid()},chat)
  return true end

function go.half(   data)
  data= csv2data("../data/auto93.csv")
  local As,Bs,_= half.splits(data.rows)
  print(#As,#Bs)
  return true end

function go.bestOrRest(   data,data1,data2,best,rest0,rest)
  data= csv2data("../data/auto93.csv")
  best,rest0 = bestOrRest(data.rows)
  rest = many(rest0, #best*4)
  data1=clone(data, best)
  data2=clone(data, rest)
  map({stats(data1),stats(data2)},chat)
  return true end

function go.bins()
  local data= csv2data("../data/auto93.csv")
  local best,rest0 = half.splits(data.rows)
  local rest = many(rest0, #best*2*the.Balance)
  local rows ={}
  for _,row in pairs(rest) do push(rows,row).label="rest" end
  for _,row in pairs(best) do push(rows,row).label="best" end
  for _,col in pairs(data.about.x) do
    print("")
    map(bins.find(rows,col),
        function(xy) print(xy.txt,xy.xlo,xy.xhi, cat(xy.y.has)) end) end
  return true end

local _ranked=function(data)
   for n,row in pairs(sort(data.rows)) do row.rank= rnd(100*n/#data.rows,0); end
   for _,row in pairs(data.rows) do row.evaled=false end
   shuffle(data.rows)
   return data  end

function go.rules(      data)
  local data= _ranked(csv2data("../data/auto93.csv"))
  how.rules(data)
  chat(sort(map(data.rows,get"rank")))
  return true end

---- ---- ---- ---- Start-up
main(help,the,go)
```